

# Тема 1 Понятие данных и информации

[1. Понятие данных и информации](#)

[2 Операции с данными](#)

[3 Виды и типы данных](#)

[4 Кодирование данных двоичным кодом](#)

[5 Основные структуры данных](#)

[6 Единицы представления, измерения, хранения и передачи данных](#)

## 1. Понятие данных и информации

Всё, что нас окружает, и с чем мы ежедневно сталкиваемся, относится либо к физическим телам, либо к физическим полям. Известно, что все физические объекты находятся в состоянии непрерывного движения и изменения, которое сопровождается обменом энергией и её переходом из одной формы в другую. Все виды энергообмена сопровождаются появлением сигналов, то есть, все сигналы имеют в своей основе материальную энергетическую природу. При взаимодействии сигналов с физическими телами в физических телах возникают определённые изменения свойств – это явление называется регистрацией сигналов. Такие изменения можно наблюдать, измерять или фиксировать иными способами. При этом возникают и регистрируются новые сигналы, то есть, образуются данные.

**Итак, данные – это зарегистрированные сигналы.**

Обработка данных адекватными им методами создаёт новый продукт – информацию. Таким образом, информация возникает и существует в момент взаимодействия объективных данных и субъективных методов. Как и всякий объект, она обладает свойствами (объекты различимы по своим свойствам). Характерной особенностью информации, отличающей её от других объектов природы и общества, является то, что на свойства информации влияют как свойства данных, составляющих её содержательную часть, так и свойства методов, взаимодействующих с данными в ходе информационного процесса. По окончании процесса свойства информации переносятся на свойства новых данных, то есть свойства методов могут переходить на свойства данных.

**Итак, информация – это продукт взаимодействия данных и адекватных им методов.**

## 2 Операции с данными

В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов. Обработка данных включает в себя множество различных операций. В структуре возможных операций с данными можно выделить следующие операции:

- сбор данных – накопление информации с целью обеспечения достаточной полноты для принятия решения;
- форматизация данных – приведение данных, поступающих из разных источников, к одинаковой форме, чтобы сделать их сопоставимыми между собой, то есть повысить их уровень доступности;
- фильтрация данных – отсеивание тех данных, в которых нет необходимости для принятия решения; при этом должны возрасти достоверность и адекватность информации;
- сортировка данных – упорядочение данных по заданному признаку с целью удобства их использования; при этом должна повышаться доступность информации;
- архивация данных – организация хранения данных в удобной и легкодоступной форме; служит для снижения экономических затрат по хранению данных и повышает общую надёжность информационного процесса в целом;
- преобразование данных – перевод данных из одной формы в другую или из одной структуры в другую;
- защита данных – комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных;
- транспортировка данных – приём и передача данных между удалёнными участниками информационного процесса.

### **3 Виды и типы данных**

Данные могут быть представлены следующими видами:

- целыми и действительными числами;
- текстом;
- мультимедийными (графическими объектами, звуковыми сигналами, цветными изображениями).

В зависимости от вида данных, они могут подразделяться на следующие типы:

- байтовый тип;
- целочисленные типы простой и двойной точности;
- типы действительных чисел простой и двойной точности;
- типы даты и времени;
- строковый тип;
- логический тип;
- тип объектов.

## 4 Кодирование данных двоичным кодом

Для автоматизации работы с данными, относящимися к различным типам, очень важно унифицировать их форму представления – для этого используется приём кодирования, то есть выражение данных одного типа через данные другого типа. В вычислительной технике применяется система кодирования двоичным кодом. Она основана на представлении данных последовательностью всего двух знаков **0** и **1**. Эти знаки называются двоичными цифрами, каждая из которых представляет 1 бит. Одним битом могут быть выражены два понятия: **0** или **1** (да или нет, чёрное или белое, истина или ложь и т. п.). Двумя битами можно выразить четыре различных понятия: **00, 01, 10, 11**.

Тремя битами можно закодировать 8 различных значений: **000, 001, 010, 011, 100, 101, 110, 111**.

Увеличивая на единицу количество разрядов в системе двоичного кодирования, в два раза увеличивается количество кодируемых значений. Общая формула расчёта имеет вид:  $N = 2^m$ ,

где **N** – количество независимых кодируемых значений;

**m** – количество разрядов двоичного кодирования.

### 4.1 Кодирование целых и действительных чисел

Двоичный код целого числа можно получить путём деления числа на 2 до тех пор, пока частное не будет равно 1. Совокупность остатков от каждого деления, записанная справа налево вместе с последним частным, и образует двоичный аналог десятичного целого числа.

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). Для кодирования чисел от 0 до 65535 потребуется 16 разрядов (16 бит). Используя 24 разряда (24 бита), можно закодировать более 16,5 миллионов разных значений.

Для кодирования действительных чисел используется 80 разрядов (80 бит). При этом действительное число предварительно преобразуется в нормализованную форму:  $41,2346785 = 0,412346785 * 10^2$ . Первая часть нормализованного числа называется мантиссой, а вторая – характеристикой. При этом значительная часть из 80 бит задействуется для хранения мантиссы (вместе со знаком числа) и некоторое фиксированное количество бит отводится для хранения характеристики (тоже со знаком степени).

### 4.2 Кодирование текстовых данных

Если каждому символу присвоить порядковый номер (целое число), то с помощью двоичного кода можно кодировать любые текстовые данные. Восемью двоичных разрядов достаточно для кодирования 256 различных символов. Этого хватит, чтобы закодировать комбинациями 8 битов все символы английского и русского алфавитов (строчные и прописные), арабские цифры,

знаки препинания, символы арифметических действий и некоторые общепринятые специальные символы.

С этой целью институт стандартизации США (ANSI – American National Standard Institute) ввёл в действие систему кодирования ASCII (American Standard Code for Information Interchange – стандартный код информационного обмена США). В системе ASCII закреплены две таблицы кодирования – базовая и расширенная. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 по 255.

Первые 32 кода (от 0 до 31) базовой таблицы выделены производителям аппаратных средств (в первую очередь компьютеров и печатающих устройств). Это управляющие коды, которым не соответствуют никакие символы, ими можно управлять работой технических устройств.

Коды от 32 по 127 предназначены для кодирования символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов. Символы русского алфавита и другие специальные символы кодируются кодами расширенной таблицы от 128 по 255.

Однако, рассмотренная выше система кодирования ASCII, не обеспечивает кодирование алфавитов многих других языков планеты. С целью устранения этого недостатка в настоящее время создана универсальная система – UNICODE, основанная на 16 – разрядном кодировании символов. Эта система позволяет обеспечить уникальные коды для 65536 различных символов. Этого количества достаточно для размещения в одной таблице символов большинства различных алфавитов планеты.

### 4.3 Кодирование графических данных

Если графическое изображение рассматривать как комбинацию мельчайших точек, образующих определённый узор, называемый растром. То с помощью линейных координат и индивидуальных свойств каждой точки, выраженных с помощью целых чисел, можно применить систему двоичного кодирования и для графических данных. К индивидуальным свойствам точки относятся яркость и цвет.

Чёрно – белые иллюстрации представляются в виде комбинации точек с 256 градациями серого цвета. Таким образом, для кодирования яркости любой точки достаточно 8 разрядов двоичного числа.

Кодирование цветных графических изображений осуществляется на принципе декомпозиции произвольного цвета на основные составляющие. В качестве таких составляющих используются три цвета: красный (*Red, R*), зелёный (*Green, G*) и синий (*Blue, B*). Такое кодирование называется системой *RGB*. При этом если для кодирования яркости каждой из основных составляющих использовать по 256 значений (8 двоичных разрядов), то на кодирование цвета одной точки требуется 24 разряда. Такая система кодирования обеспечивает 16,5 миллионов цветов. Эта система является полноцветной и называется *True Color*. Если уменьшить количество двоичных разрядов, используемых для кодирования цвета каждой точки, то

можно сократить объём данных, но при этом заметно сократится диапазон кодируемых цветов. Кодирование цветной графики двоичными числами, содержащими 16 разрядов, называется *High Color*.

На практике применяется индексный метод кодирования информации о цвете. При этом код каждой точки растра выражает не цвет сам по себе, а только его номер (индекс) в справочной таблице, называемой *палитрой*, которая прилагается к графическим данным.

#### 4.4 Кодирование звука

Для кодирования звуковой информации применяется метод таблично – волнового синтеза (*Wave– Table*). Сущность этого метода состоит в том, что используются заранее подготовленные таблицы образцов звуков. В технике такие образцы называют *сэмплами*. Числовые коды звуковой информации выражают: тип инструмента и номер его модели, высоту тона, продолжительность, интенсивность звука и динамику его изменения. А также некоторые параметры среды, в которой происходит звучание, и прочие параметры, характеризующие особенности звука.

### 5 Основные структуры данных

Работа с большими наборами данных автоматизируется проще, когда данные упорядочены, то есть образуют заданную структуру. Существует три основных типа структур данных: **линейная, табличная и иерархическая**. При создании любой структуры данных необходимо обеспечить решение двух задач: как разделять элементы данных между собой и как разыскивать нужные элементы.

Линейные структуры – это хорошо знакомые списки. Список – это простейшая структура данных, отличающаяся тем, что каждый элемент данных однозначно определяется своим уникальным номером в массиве (списке).

Табличные структуры данных подразделяются на двумерные и многомерные.

Двумерные табличные структуры данных (матрицы) – это упорядоченные структуры, в которых адрес элемента определяется номером столбца и номером строки, на пересечении которых находится ячейка, содержащая искомый элемент.

Многомерные таблицы – это упорядоченные структуры данных, в которых адрес элемента определяется тремя и более измерениями. Для отыскания нужного элемента в таких таблицах необходимо знать параметры всех измерений (размерностей).

Линейные и табличные структуры являются простыми. Ими легко пользоваться, поскольку адрес каждого элемента задаётся числом (для списка), двумя числами (для двумерной таблицы) или несколькими числами для многомерной таблицы. Они также легко упорядочиваются. Основным методом

упорядочения таких данных является сортировка. Недостатком простых структур данных является трудность их обновления. При добавлении, например, произвольного элемента в упорядоченную структуру возникает необходимость изменения адресных данных у других элементов.

Иерархические структуры – это структуры, объединяющие нерегулярные данные, которые трудно представить в виде списка или таблицы. В иерархической структуре адрес каждого элемента определяется маршрутом, ведущим от вершины структуры к данному элементу. Эти структуры по форме сложнее, чем линейные и табличные, но они не создают проблем с обновлением данных. Их легко развивать путём создания новых уровней. Недостатком иерархических структур является относительная трудоёмкость записи адреса элемента данных и сложность упорядочения. Поэтому для упорядочения в таких структурах применяется метод предварительной индексации. При этом каждому элементу данных присваивается свой уникальный индекс, который используется при поиске, сортировке и тому подобное. В качестве примера иерархической структуры может служить система почтовых адресов.

## **6 Единицы представления, измерения, хранения и передачи данных**

Одной из систем представления данных, принятых в информатике и вычислительной технике является система двоичного кодирования. Наименьшей единицей такого представления является бит (двоичный разряд).

Совокупность двоичных разрядов, выражающих числовые или иные данные, образует некий битовый рисунок. С битовым представлением удобнее работать, если этот рисунок имеет регулярную форму. В качестве таких форм используются группы из 8 битов, каждая из которых называется **байтом**. Однако во многих случаях целесообразно использовать 16 – разрядное, 24 – разрядное, 32 – разрядное, 64 – разрядное кодирование.

Байт является наименьшей единицей измерения количества данных (информации).

Более крупные единицы измерения данных образуются добавлением префиксов кило-, мега-, гига-, тера-.

$$1 \text{ Килобайт (Кбайт)} = 1024 \text{ байт} = 2^{10} \text{ байт.}$$

$$1 \text{ Мегабайт (Мбайт)} = 1024 \text{ Кбайт} = 2^{20} \text{ байт.}$$

$$1 \text{ Гигабайт (Гбайт)} = 1024 \text{ Мбайт} = 2^{30} \text{ байт.}$$

$$1 \text{ Терабайт (Тбайт)} = 1024 \text{ Гбайт} = 2^{40} \text{ байт.}$$

В более крупных единицах пока нет практической надобности.

В качестве единицы хранения данных (информации) принят объект переменной величины, называемый файлом

**Файл – это последовательность произвольного числа байтов, обладающая уникальным собственным именем.**

Поскольку в определении файла нет ограничений на его размер, то можно представить себе файл, имеющий 0 байтов (пустой файл), и файл, имеющий

любое число байтов. В определении файла особое внимание уделяется имени. Имя файла фактически несёт в себе адресные данные, без которых, данные, хранящиеся в файле, не станут информацией из-за отсутствия методов доступа к ним. Кроме адресных функций, имя файла может хранить сведения о типе данных, заключённых в нём.

Требование уникальности имени файла в вычислительной технике обеспечивается автоматически – создать файл с именем, тождественным уже существующему, не может ни пользователь, ни автоматика. Уникальность имени файла обеспечивается тем, что полным именем файла считается собственное имя файла вместе с путём доступа к нему.

Хранение файлов организуется в иерархической структуре, которая называется файловой структурой. В качестве вершины структуры служит имя носителя, на котором сохраняются файлы. Далее файлы группируются в каталоги (папки), внутри которых могут быть созданы вложенные каталоги (папки). Путь доступа к файлу начинается с имени устройства и включает все имена каталогов (папок), через которые проходит. В качестве разделителя используется символ “\” (обратная косая черта).

Синтаксис записи полного имени файла:

*Имя носителя \ Имя каталога 1 \ Имя каталога N \ Собственное имя файла .*

Пример: *C\Игры\Стрелялки\Кролики.*

Передача данных в компьютерных системах измеряется её скоростью. Единицей измерения скорости передачи данных через последовательные порты является: бит в секунду (бит/с, Кбит/с, Мбит/с). Единицей измерения скорости передачи данных через параллельные порты является байт в секунду (байт/с, Кбайт/с, Мбайт/с).

### **Вопросы для самоконтроля**

1. В чём состоит отличие между данными и информацией (понятие данных, понятие информации)?
2. Какие основные операции можно осуществлять с данными?
3. Какие достоинства и недостатки присущи основным структурам данных?
4. Как осуществляется кодирование двоичным кодом?
5. Что собой представляет система кодирования ASCII (базовая и расширенная таблицы кодов).
6. Какая современная система кодирования обеспечивает кодирование большинства алфавитов планеты?
7. Как называются единицы представления, измерения, хранения и передачи данных? Какова их размерность?